PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of:	)	**Group Art Unit: 2137**
	)	
Harald VATER *et al.*	)	**Examiner: Z. Davis**
	)	
Serial Number: 09/700,656	)	***Attorney Docket*: VATE3001/BEU**
	)	
Filed: February 14, 2001	)	**Confirmation No.: 2137**

For: Access-Controlled Data Storage Medium

APPELLANT'S REPLY BRIEF UNDER 37 C.F.R. §41.41

Sir:

This paper is a Reply Brief in furtherance of the Notice of Appeal and Appeal Brief filed in this case on April 19, 2011, and is responsive to the Examiner's Answer dated July 8, 2011.

This Brief addresses the following specific errors or misstatements included in the Examiner's Answer, under the following headings:

1. Summary of Appellant's Arguments Made in Appeal Brief
2. Error #1: Cordery is "Suggestive of Pre-Computing and Storing for Later Use Any Number of Security Relevant Values"
3. Error #2: Kocher's Auxiliary Data and Auxiliary Function Values are "Security-related Function Values" of the Type Suggested to be Pre-Computed by Cordery
4. Error #3: The Timing of the Permutations of Kocher is Not Critical
5. Error #4: The Argument Concerning the Changes to Kocher that are Necessary to Obtain the Claimed Invention are "Largely Conjecture"

1. Summary of Appellant's Arguments Made in Appeal Brief

Appellant's argument may be summarized as follows:

- The Kocher publication discloses disguising of input data by combining the input data with auxiliary data (Z), performing operations (f) on the disguised input data, and then compensating for the use of disguised input data by combining the resulting output data with

auxiliary function values ($f(Z)$). The auxiliary function values ($f(Z)$) are calculated at the time that the operations are performed.

- Cordery teaches pre-computation of digital tokens to, at least in part, protect secret input data by performing *all* of the computations previously in a secure environment. In doing so, Cordery's method eliminates any need for disguising data as taught by Kocher. On the other hand, by disguising the input data, Kocher allows the operations to be performed in an *insecure* environment, making it unnecessary to performs operations previously in a safe environment. As far as can be determined from the disclosure of Kocher, the input data is completely protected and cannot be discovered and there is no possible need to even generally apply the teachings of Cordery, much less to selectively apply the teachings of Cordery to one particular aspect of the method of Kocher, namely the calculation of auxiliary function values.
- The method of Kocher does have a weakness, but the weakness is taught only by the present application and is not apparent from Kocher or Cordery. The weakness is that the nature of the operations (f) and therefore the manner in which the auxiliary functions values are determined can possibly be determined by statistical analysis of power transmissions from the chip that performs the operations, if a sufficiently large number of operations are observed. Kocher is not concerned with such statistical analyses, and provides no hint that its contemporaneous computation of auxiliary function values is in any way vulnerable.
- The claimed invention overcomes the weakness of the method of Kocher (which weakness is not recognized by Kocher), by pre-computing the auxiliary function values and auxiliary data in safe surroundings and storing them in a memory on the chip, rather than calculating the auxiliary function values at the time the operations on the falsified input data are performed. In other words, where Kocher differs from the claimed invention is that the functions used to disguise the operations in the insecure environment are calculated at the time the operations are performed, whereas the claimed invention pre-computes those functions. Kocher provides no reason for pre-computing the auxiliary function values since auxiliary function values are used solely to compensate for the disguised input data, which does not affect the disguising of the input data or the results of the computations.

2. Error #1: Cordery is "Suggestive of Pre-Computing and Storing for Later Use Any Number of Security Relevant Values" (page 10, lines 2-3 of Examiner's Answer)

The Appellant argues in the Appeal Brief that, in order to obtain the claimed invention, one of ordinary skill in the art would have needed to disregard or ignore the extensive and specific teachings in Kocher concerning the manner in which the auxiliary function values are calculated while the operations on the disguised input data are being performed. In reply, the Examiner's position is that Cordery teaches such as modification because, in Cordery, certain secret values are pre-computed, and because the Cordery and Kocher publications both generally concerned with "security-related values." As explained in the first paragraph on page 10 of the Examiner's Answer:

> *. . . as broadly interpreted, Cordery is suggestive of pre-computing and storing for later use any number of security-related values. . .which are considered to include the auxiliary function values as claimed.*

In other words, according to the Examiner, if one "interprets" the teachings of Cordery broadly enough, they not only suggest pre-computation of digital tokens, but also pre-computation of any "security-related" value, even if the "security-related" value is a value, as taught by Kocher, that is re-calculated *each* time operations are performed on already-disguised input data. However, the Examiner's reply provides no explanation as to why Cordery's pre-computation of digital tokens in order to protect "security-related" data should be applied to a value that is re-calculated each time a procedure on already-disguised data is performed?

Cordery's disclosure provides a specific solution to the problem of key updating and managing in a postage payment system (see col. 2, lines 42-44 and 59-61 of Cordery). The solution is to pre-compute a limited number of digital tokens and store them on a secure data carrier. Because the data items are pre-computed, it is no longer necessary to execute encryption on the data carrier itself and therefore, it is also no longer necessary to store secret keys that are used for encryption on the data carrier. As explained in col. 3, lines 18-31 of Cordery, pre-computing of a limited number of digital tokens and storing them on a secure data carrier reduces the number of secret keys that must be managed, thereby making key management and updating for efficient. The question is, what does this teaching have to do with Kocher? The Examiner provides no explanation. **In particular, the Examiner does not explain why Cordery's teaching of reducing the number of secret keys (by pre-computation and secure storage of certain digital tokens)**

should be applied to Kocher's teaching of calculating auxiliary function values during the performance of operations on disguised secret data. Are there any advantages to such a modification of Kocher's method? The references do not provide any.

An advantage of modifying Kocher's method to include pre-computation and storage of auxiliary function values does exist, but it is not suggested by either Kocher or Cordery. It is only suggested by the present application. The advantage has to do with statistical power analysis. This advantage has nothing to do with reducing the number of keys as taught by Cordery. The digital tokens of Cordery are not auxiliary function values that compensate for the disguising of input data, and Cordery does not suggest any vulnerability in the method of Kocher that would have caused the ordinary artisan to ignore the teachings of Kocher concerning contemporaneous generation of the auxiliary function values used to compensate for input data falsification and instead to pre-compute of Kocher's auxiliary function values. **Instead, the sole basis offered by the Examiner for modifying Kocher in view of Cordery is apparently that Kocher and Cordery both happen be "*concerned with the encryption and protection of secret data*" (last paragraph on page 10 of the Official Action), thereby making the references "analogous" and eliminating the need for the Examiner to consider the specific teachings of the references.**

The fact that Kocher protects its pre-computed input data by disguising it, but teaches contemporaneous computations of auxiliary function values that compensate for the disguising of secret input data is apparently irrelevant to the Examiner, as is the Cordery's complete lack of any teachings that even remotely suggest that there is any problem with contemporaneous generation of auxiliary function values. Kocher does use pre-computed keys and values (the input data), and possibly might protect them by hardware in the manner of Cordery. But Kocher then goes way beyond the teachings of Cordery by disguising the secret values after they have been retrieved, using a detailed and complex algorithm involving both auxiliary data to disguise the secret values and auxiliary function values to compensate for the use of the auxiliary data to disguise the secret values. Absolutely nothing in Cordery is even remotely directed to protection of auxiliary function values generated for the purpose of compensating for the use of auxiliary data to protect secret input data.

Cordery is only concerned with the secret input data, and not with disguising of the secret input data, much less with auxiliary function data that is generated at the time operations are performed on the disguised secret input data. As a result, the teachings of Cordery are far from being as broad and general as alleged by the Examiner, but rather are directed to a specific solution to a specific problem, namely reducing the number of secret keys be pre-computing digital tokens, which is not even remotely pertinent to the problem of compensating for use of auxiliary data to disguise input data during performance of operations on the disguised data, as taught in the Kocher publication.

Both Kocher and the claimed invention are directed to the problem of carrying out operations on secret data in an insecure environment. For example, monetary transactions and access control, are operations that must be carried out in an insecure environment. The usual solution is to perform the operations on a chip, but the claimed invention and Kocher both recognize the vulnerability of detecting operations occurring on the chip by detecting radiation from the chip. In contrast, Cordery teaches that the whole problem of protecting operations being carried out in an insecure environment can be avoided by simply performing the operations in a secure environment. This eliminates the need for *any* of the measures taught by Kocher, but the solution offered by Cordery can only be used in certain circumstances, and is not applicable to the method of Kocher, which is specifically designed to be used in an unsafe or insecure environment. Some operations must be performed in an insecure environment, such as verifying a credit card payment or opening an access gate, and Kocher addresses the problem of protecting those operations. In other words, Kocher deals with operations that must be performed in the insecure environment, and teaches a method for doing so in a more secure manner by disguising the input data using auxiliary data and, during performance of the operations, calculating auxiliary function values that can be used to compensate for the disguised input data. **If the operations of Kocher could be pre-performed in the manner of the digital token computations of Cordery, then there would be no need for disguising the input data or calculating auxiliary function values in the first place**,

Thus, the applicability of the "general concept" allegedly taught by Cordery, when "interpreted" in the broadest sense, namely to not perform encryption operations on a data carrier

in an insecure environment, ends where the problems addressed by Kocher start, namely where is no longer possible to avoid executing operations on security relevant data in an insecure environment, necessitating disguising of the data used in the operations as they are performed out in the open, rather than simply moving the operations "inside" to a secure environment. The claimed invention then starts where Kocher ends, namely with the problem of repeated performance of the operations in the insecure environment, with the result that even disguised data might be discovered through statistical power analysis techniques if enough operations are observed over a sufficient period of time. Put another way, since all of Cordery's token generating operations are performed in a secure environment, there is absolutely no need to disguise input data of Kocher, and therefore no need to generate auxiliary function values to compensate for the disguised input data. Since Cordery's method eliminates any need for generating auxiliary function values to compensate for disguised input data, Cordery could not possibly be suggestive of modifying the way in which those auxiliary function values are generated by Kocher.

Paradoxically, what the Examiner has in effect done is to **generalize** the specific teachings of Cordery concerning pre-computing of digital tokens for a postage meter to apply to any secret data, irrespective of context, and then to selectively apply the hypothetical "general" teaching to a very **specific context** that has no disclosed or logical relationship to the specific context of Cordery, namely Kocher's blinding and unblinding vectors. Furthermore, this is done without any attempt to explain why one of ordinary skill in the art would have applied the general teaching to specific values that are only used to compensate for data that would not even need to be disguised if the data were pre-computed in the first place as taught by Cordery.

3. Error #2: Kocher's Auxiliary Data and Auxiliary Function Values are "Security-related Function Values" of the Type Suggested to be Pre-Computed by Cordery (lines 1-4 on page 18 of Examiner's Answer)

What Cordery and Kocher seek to protect are secret data, *i.e.*, keys and sensitive data, and operations performed on the secret data, such as encryption and decryption operations. Cordery protects the data and operations by performing them in a secure environment, while Kocher protects the data and operations by disguising them so that they can be performed in an insecure

environment, such as a point of sale. However, nobody seeks to protect the values that happen to be generated as part of the data and operation disguising procedures, *i.e.*, the auxiliary function values of Kocher are simply not recognized to be "security-related" in a manner that would cause the ordinary artisan to apply the teachings of Cordery in the manner suggested by the Examiner. It is the key insight of the present invention that this data needs to be protected too, even though they are generated only during the disguising operations and never used again.

Even if Cordery's teachings could be extrapolated to render obvious the pre-computation of all secret data, in any environment no matter the context, including secret input data that is disguised using auxiliary data and auxiliary function values as taught by Kocher, there is nothing in either Cordery or Kocher that the auxiliary data and auxiliary function values generated during the disguising operations of Kocher correspond to the secret input data pre-computed by Cordery. The auxiliary data and auxiliary function values of Kocher are used to protect secret input data–they are not the secret data to be protected. This is why Kocher takes no special steps to protect the auxiliary function values–the auxiliary function values are generated to protect the secret data rather than being secret data to be protected. If Kocher does not take steps to protect the auxiliary function values used to protect the secret input data and operations performed thereon, why would the ordinary artisan? Cordery might teach protection of secret input data, but there is no teaching whatsoever, in any reference, that the auxiliary function values of Kocher are secret input data of the type protected by Cordery. Instead, it is the data and operations disguised by the auxiliary data and function values that constitutes secret data to be protected, and not the auxiliary data and function values generated during the disguising process.

As explained in the Appeal Brief, Kocher computes a blinding vector "b" based on a randomized bit order permutation table or array "perm," which corresponds to the claimed auxiliary function value. According to paragraph [0071] of the Kocher publication:

> **Because the process of constructing the bit order table does not involve any secret inputs, the only security requirement for the process is that the final result be unknown to attackers** (emphasis added).

Thus, Kocher clearly does not recognize that the auxiliary function value is secret data to be protected. Since Kocher explicitly teaches that the auxiliary function value is in fact data that "**does not involve any secret inputs**" and does not need to be protected, there is absolutely no logical reason to apply the secret value protection teachings of Cordery or any other reference, no matter how broadly "interpreted," to the auxiliary function values of Kocher. This is, in fact, a specific teaching away from the combination proposed by the Examiner. It is only in hindsight that anyone of ordinary skill in the art could possibly have recognized that the Kocher's auxiliary function values are "security-related data" to which the teachings of Cordery are alleged by the Examiner to apply. There is not a single teaching in either Kocher or Cordery that would have caused the ordinary artisan to recognize the necessity to also protect the auxiliary data/function value of Kocher by pre-computing the data, as claimed.

In the last sentence on page 13 of the Examiner's Answer, the Examiner appears to justify the failure to cite any teachings that would have justified the claimed modification of Kocher by stating that "***there is nothing in Kocher that explicitly prevents such a value from being pre-computed***." In other words, even though the prior art in no way teaches or suggests pre-computing of Kocher's auxiliary function values, but to the contrary explicitly states in paragraph [0072] that the auxiliary function values are not even secret data that needs to be protected, the Examiner still considers to be combination to be obvious because there is "nothing to prevent" the modification. According to the Examiner, the entire process of generating auxiliary function values taught by Kocher can simply be ignored, because there is "nothing to prevent" one from ignoring the process and instead substituting a process that is not disclosed or suggested, explicitly or implicitly, by any of the references of record. This position stands *Graham v. Deere* on its head and is absurd.

4. Error #3: The Timing of the Permutations of Kocher is Not Critical

The Examiner has apparently misunderstood the timing of the permutations of Kocher, as evidenced by the paragraph bridging pages 14 and 15 of the Examiner's Answer. According to the Examiner, Kocher (paragraph [0074]) teaches that "blinding can occur before additional permutations (corresponding to the claimed executed operations) take place." In making this

argument, the Examiner is attempting to imply that the order in which the steps data blinding and auxiliary function value compensation steps are performed is not critical, and that one of ordinary skill in the art could freely rearrange the steps in order to pre-compute the auxiliary function values. A detailed consideration of the teachings of Kocher concerning the permutation "perm" reveals that this is not the case. The permutation *perm* cannot be performed after the blinding step of Kocher.

According to Kocher, the permutation *perm* is used to create operation order entropy, *i.e.*, to alter in an unforeseeable manner the order in which the items of the secret data arrays "DataIn" and "DataOut" are accessed during further operations, in particular during the blinding step and during the permutation according to the array "table" (see paragraphs [0066] to [0068] of Kocher). As can already be seen from the respective chapter heading (paragraph [0064] of Kocher), the main method for protecting the secret data "DataIn" and "DataOut" is not blinding the data, but introducing the concept of "Execution Path and Operation Order Entropy." Blinding is only marginally mentioned as a technique to further conceal the secret data (see, the last sentence of paragraph [0068] of Kocher). In other words, performing the permutation *perm*, *i.e.*, creating operation order entropy, *after* the blinding step would result in a loss of security in the method of Kocher, since the blinding operation would then be executed in some standard order, which renders information leakage possible, as explained in paragraph [0068]. Consequently, the ordinary artisan would certainly not even think about interchanging these steps in order to permit pre-computation of the auxiliary function values.

5. Error #4: The Argument Concerning the Changes to Kocher that are Necessary to Obtain the Claimed Invention are "Largely Conjecture" (lines 7-10 on page 14 of Examiner's Answer)

Finally, the Appellant has argued that the difference between the claimed invention and the method of Kocher is that Kocher teaches contemporaneous calculation of auxiliary function values used to compensate for disguising of input data by auxiliary data. The Examiner considers this difference to be "largely conjecture" (lines 7-10 on page 14 of the Examiner's Answer) The Appellant disagrees. The argument concerning the differences between the claimed invention and Kocher is based on the recitation in claim 26 that "*the auxiliary function value (f(Z)) was* ***previously determined*** *by execution of the one or more operations (f) with the auxiliary data (Z) as input data*

in safe surroundings and ***stored*** *along with the auxiliary data (Z) in the memory of the semiconductor chip of the data carrier*," which is contrasted with Kocher's teachings that calculation of the auxiliary value during performance of the algorithm. Indeed, the Examiner states in the last complete sentence on page 13 of the Examiner's Answer that "it is acknowledged that the described implementation of the method of Kocher does disclose calculation of an auxiliary value during performance of the algorithm." There is nothing conjectural about this statement of the difference.

More specifically, the Examiner considers the following statements of what the ordinary artisan would have had to recognize in order to modify the method of Kocher in the manner proposed by the Examiner to be "largely conjecture":

a. the ordinary artisan would have had to recognize that the method of Kocher may, at least in principle and despite explicit teachings to the contrary, be changed without any loss of security and without changing the output values by pre-computing the random bits *b and* the random permutation *perm* so that *b* and *perm* would serve as input data in addition to the actual input data *dataIn*, *dataOut*, and *table* (which would require considerable algorithmic skills not even remotely taught by Cordery);

b. the ordinary artisan would have had to further recognize that the blinding bits *b* would need to be pre-computed in safe surroundings and stored in an array of random blinding bits, for simplicity also called *b*, and that the random permutation *perm* would also have to be pre-computed in safe surroundings;

c. the ordinary artisan would have had to further recognize that the unblinding vector, stored in the vector *dataOut*, would have had to be pre-computed by applying the permutation *perm* and the permutation *table*, in that order, to the random vector *b*, *i.e.*, *dataOut [table [perm[i]] := b[i]*; and

d. the ordinary artisan would have had to provide for storing the random vector *b* representing the auxiliary data along with the unblinding vector *dataOut* representing the auxiliary function value, with the result that the main routine to compute the actual permutation of the input array *dataIn* according to the array *table* would then comprise the following blinding steps:

```
for (i=1; i<64; i++){
p=perm[i];                    //perm has already been pre-computed
temp [p] := dataIn[p] ^ b[i];          //random vector b[i] has
                                       // already been pre-computed}
```

Which of these statements is "largely conjecture"? How could pre-computing of the auxiliary data and function values as claimed be achieved without pre-computing the values that are necessary to

compute the auxiliary data and function values? It is not conjecture to argue that pre-computing of the auxiliary function values, as claimed, also requires pre-computing of all of the variables that are used to compute the auxiliary function values. This is simple logic that follows from the teachings of Kocher. What is conjecture is that one of ordinary skill in the art would have "interpreted" Cordery's teachings concerning pre-computation of digital tokens to apply to the bits *b*, permutation *perm*, and so forth used to compute auxiliary function values in the context of Kocher's data and operation disguising method, when neither Cordery nor Kocher provides any reason whatsoever for making such a modification.

Conclusion

For all of the foregoing reasons, and for the reasons given in the Appeal Brief submitted on April 19, 2011, Appellants respectfully submit that the Examiner's final rejection of claims 26-33 and 42 under 35 U.S.C. §103(a) is improper and should be reversed by this Honorable Board.

Respectfully submitted,

BACON & THOMAS, PLLC

/Benjamin E. Urcia/

Date: September 8, 2011 By: BENJAMIN E. URCIA
Registration No. 33,805

BACON & THOMAS
625 Slaters Lane, 4th Floor
Alexandria, Virginia 22314

Telephone: (703) 683-0500

S:\Producer\beu\Pending Q...Z\V\VATER 700656\ReplyBrief.wpd